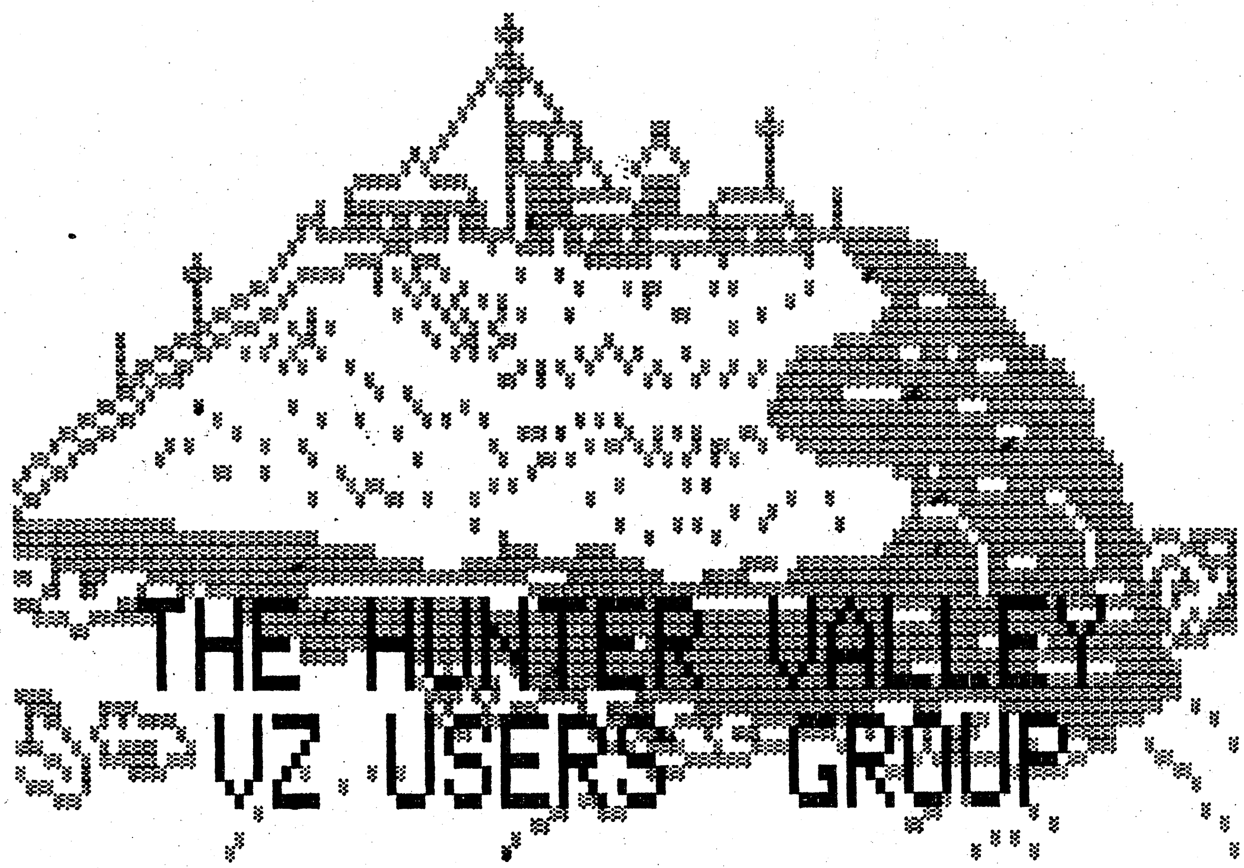
MAY/JUNE 1988 ISSUE # 18




# VZ 200/300

# HUNTER VALLEY VZ JOURNAL

PRODUCED BI-MONTHLY BY H.V.VZ.U.G.

FRONT COVER :-
Novocastrians should recognise Nobby's Lighthouse which was drawn by Matthew Taylor. To my knowledge this VZ publication is the first to feature a front cover designed using the VZ. My thanks go to Matthew for an exellent contribution.

EDITORIAL - USER GROUPS :- Page 3

MATH'S TESTER by Brooke SPONG of NZ :- Pages 4-6
Brooke presents a usefull program for the four math's functions. It caters for all ages by allowing you to select your level. A nice touch is you get comments as you proceed.

SHIFT LOCK SWITCH UPDATE by Joe LEON :- Page 6
The original SHIFT LOCK SWITCH was published in the JAN/FEB 1987 issue of Hunter Valley VZ Journal and after some comments and suggestions an updated version is presented.

This version adds power on reset so that on power up the S/Lock switch is in the off position plus Switch debouncing is incorporated as well.

ROBERT QUINN DATABASE REVIEW by Peter J.HILL NZ :- Page 7
Peter as usual does a marvelous job on writing reviews and I must agree with his comments on DATA being very useful, well written and good value for money.

24 BIT I/O P/P/INTERFACE by Bob KITCH (C) :- Pages 8-10
After a long absence Bob presents a well written and presented and easy to understand project on how to interface the VZ to the outside world.

JAGULAR JETTY by Adam MAGEE :- Pages 11-12
This game for the VZ can best be described as a single level platform game. It's not as easy as it looks. You have to overcome a few obstacles along the way and beat the clock as well.

64K RAM PACK BANK SWITCHING by Chris HOBROUGH :- Pages 13-15
Chris explains in his easy to understand way on how to use bank switching from assembly or basic. He also gives useful details on the makeup of the 64K Ram Pack. On page 14 also by chris is a Memory Monitor program to select and test writing/reading to different banks.

NMI INTERRUPT UPDATE by Joe LEON :- Page 15
The original NMI INTERUPT was designed by Dave Mitchell and published in the NOV. 1986 issue of Hunter Valley VZ Users' Group Magazine.

This updated version performs same function as before but adds synchronized NMI and RESET functions.

Both the S/Lock and NMI Interrupt projects share the same two IC'S. Cost for building both projects is about $5.00

128K SIDEWAYS RAM by Joe LEON (C) Pages 16-17
If insuficient memory is your problem then this project will solve it. It's double what the 64K Ram Pack gives you. 128K is by no means the limit as it's quite easy to take it up to 256K.

SOFTWARE FOR SALE :- Pages 18-20
Our range of Software ads is growing with ads from Dave Mitchell, Matthew Taylor Software, Vee Zed Down Under, VSOFTWAREZ and Hunter Valley VZ Users' Group. The program you need may just be in one of the ads, so don't skip them.

BELIEVE IT OR NOT :-
My son, yes him again was having trouble SAVING a program on the C64. After many failed attempts he typed in SAVE"PLEASE", pressed RETURN and it worked. Who said teenagers didn't know how to use their manners.

VZ NEWS - VZ NEWS - VZ NEWS

It's official - The VZ has been discontinued. It's no longer in D.Smith's catalogues. If you wanted any VZ gear you better grab it before existing stocks are depleted. You may even get it at a bargain price.

NEWCASTLE MICROCOMPUTING EXHIBITION :-

Hunter Valley VZ Users' Group has taken part in above annual show and we should be there again next year. Our aim was to let VZ Users know that there's support around and in that respect we have suceeded.

ULTRA GRAPHICS ADAPTOR :-

In the April issue of AEM there was a HI-RES GRAPHICS project for the VZ 200/300. It will give you 192X256 pixels, upper/lower case, 256 new characters, etc.

NOTE - It has come to me from couple of sources that there are wiring errors and P.C.B. is mirror image as well. Check with author or following issues of AEM for corections.

VZ 300 EPROM PROGRAMMER :-

This programmer appeared in MAY and JUNE issues of ETI. It will program 8K to 32K EPROMS. Many VZ owners will end up building this unit.

NEXT ISSUE :-

4K to 64K SIDEWAYS EPROM for the 6000H-67FFH range.
LEVEL METER for cassette recorders for CLOADING/CSAVING.

## VZ USER GROUPS AND PUBLICATIONS —

VZ USER MARK HARWOOD P.O.BOX 154 DURAL N.S.W. 2158

LE'VZ OOP J.C.E. D'ALTON 39 AGNES St. TOOWONG QLND. 4066
VSOFTWAREZ - SOFTWARE/HARDWARE FOR SALE

VZ DOWN UNDER SCOTT LE BRUN 59 BRENTWOOD DVE WANTIRNA 3152
SOFTWARE/HARDWARE FOR SALE

VZ-LINK - PETER J. HILL P.O.BOX 1972 C.P.O. AUCKLAND N.Z.

WAVZ - GRAEME BYWATER P.O. BOX 388, MORLEY W.A. 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178

HUNTER VALLEY VZ USERS' GROUP - P.O. BOX 161 JESMOND N.S.W. 2299
EDITOR-JOE LEON (049)51 2756 - SECRETARY-ROSS WOODS (049)71 2843

SUBSCRIPTION - H.V.VZ.JOURNAL - 6 MONTHS $9.00 - 12 MONTHS $18.00
New Zealand - 6 MONTHS $12.00 - 12 MONTHS $24.00

NEW VENUE - NEW DATES - NEW VENUE - NEW DATES - NEW VENUE

MEETINGS - FIRST FRIDAY of MONTH at JESMOND NEIGHBOURHOOD CENTRE
MORDUE PARADE - REAR STOCKLAND MALL (BIG W) JESMOND

NOTE :- When writing to any above or H.V.VZ. Users' Group for information please enclose a S.S.A.E. or NZ 2 Int. Reply Coupons.

```
2 '**********************************************************
4 '* MATH'S TESTER WRITTEN BY BROOKE SPONG (NZ) FOR VZ200/300 *
6 '**********************************************************
8 :
10 CLS
12 GOSUB30000
14 GOSUB2000
16 S=0
20 CLS:PRINT@195,"MULTIPLICATION QUESTIONS"
30 PRINT:PRINT " HERE ARE 10 MULTIPLICATION "
38 PRINT" QUESTIONS AT LEVEL ";A$;" TO        TEST YOUR SKILLS"
40 FORF=1TO3000:NEXTF
41 IFA$="L1"THENX=INT(RND(0)*7)-1:Y=RND(6):IFX<0THEN41
42 IFA$="L2"THENX=RND(12):Y=RND(12):IFX<7ORY<7THEN42
43 IFA$="L3"THENX=RND(15):Y=RND(15):IFX<10ORY<10THEN43
44 IFA$="L4"THENX=RND(50):Y=RND(50):IFX<15ORY<15THEN44
48 CLS:PRINT@76,"LEVEL"
50 PRINT@66,"SCORE":PRINT@86,"ATTEMPTS":PRINT@291,"REPORT";"
51 PRINT@99,S:PRINT@121,T:PRINT@110,A$
52 T=T+1
53 IFT=11THENPRINT@299,"PROCESSING FINAL                    REPORT"
54 IFT=11THENFORF=1TO800:NEXTF:GOTO91
59 PRINT: PRINT@200,Y" X ";X;"=";
61 INPUTZ
65 IF Z=Y*X THEN S=S+1:GOSUB1000:GOTO41
70 IF Z<>Y*X THEN GOSUB100
80 GOTO48
91 FORK=1TO800:NEXTK:CLS:PRINT@66,"#######FINAL REPORT########"
92 PRINT: PRINT@128,"YOUR SCORE WAS";S;"CORRECT OUT OF ";
93 PRINT T-1 ;"ATTEMPTS WHICH EQUALS";INT(S/(T-1)*100);"%"
94 IFINT(S/(T-1)*100)>80THENGOSUB200
95 PRINT: PRINT"DO YOU WISH TO CARRY ON?? IF SO   PRESS RETURN";
96 INPUT Y
97 CLS
98 GOTO14
100 SOUND20,4;15,6:PRINT@299,"OOP'S TRY AGAIN"
105 FORG=1TO800:NEXTG
110 RETURN
200 PRINT: PRINT"THAT'S VERY CLEVER OF YOU GETING OVER 80% FOR ";
210 PRINT"THIS LEVEL ";A$;" TEST SO GOOD IN FACT THAT";
220 PRINT" I'LL PLAY  YOU A TUNE...."
225 FORU=1TO2000:NEXTU
230 SOUND15,3;15,4;15,3;20,5;20,5;22,5;22,5;27,6;24,3;20,3;20,2
231 SOUND24,3;20,3;17,5;25,6;22,3;19,2;20,5
240 RETURN
300 T=0:S=0
305 CLS:PRINT@200,"DIVISION QUESTIONS"
310 PRINT:PRINT" HERE ARE 10 DIVISION QUESTIONS  AT LEVEL ";A$;
320 PRINT" TO TEST YOUR SKILLS":FORF=1TO3000:NEXTF
350 IFA$="L1"THENY=RND(5):X=RND(5)*Y
351 IFA$="L2"THENY=RND(12):X=RND(12)*Y:IFY<6THEN351
352 IFA$="L3"THENY=RND(15):X=RND(15)*Y:IFY<10THEN352
353 IFA$="L4"THENY=RND(50):X=RND(50)*Y:IFY<15THEN353
360 CLS:PRINT@76,"LEVEL"
365 PRINT@66,"SCORE":PRINT@86,"ATTEMPTS":PRINT@291,"REPORT"
370 PRINT@99,S:PRINT@121,T:PRINT@110,A$
371 T=T+1
372 IFT=11THENPRINT@299,"PROCESSING FINAL                    REPORT"
373 IFT=11THENFOR F=1TO800:NEXTF:GOTO91
375 PRINT:PRINT@200,X" % ";Y;"=";
380 INPUT Z
390 IFZ=X/YTHENS=S+1:GOSUB1000:GOTO350
395 IFZ<>X/YTHENGOSUB100:GOTO360
```

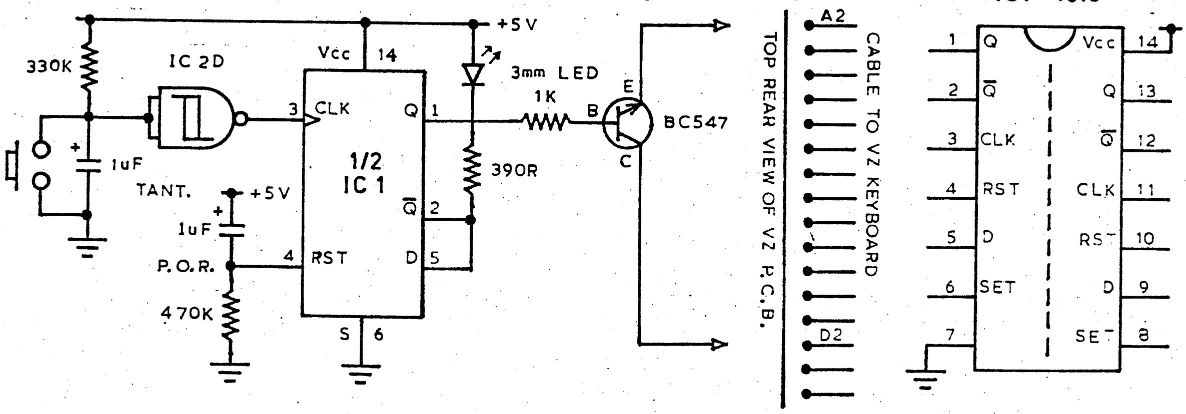
```
400 CLS:PRINT@198,"ADDITION QUESTIONS"
410 PRINT:PRINT" HERE ARE 10 ADDITION QUESTIONS  AT LEVEL ";A$;
420 PRINT" TO TEST YOUR SKILLS":FORF=1TO3000:NEXTF
430 T=0:S=0
431 IFA$="L1"THENY=RND(15):X=RND(15)
432 IFA$="L2"THENY=RND(50):X=RND(50):IFX<15ORY<15THEN432
433 IFA$="L3"THENY=RND(200):X=RND(200):IFX<50ORY<50THEN433
434 IFA$="L4"THENY=RND(2000):X=RND(2000):IFX<200ORY<200THEN434
445 CLS:PRINT@76,"LEVEL"
450 PRINT@66,"SCORE":PRINT@86,"ATTEMPTS":PRINT@291,"REPORT;"
460 PRINT@99,S:PRINT@121,T:PRINT@110,A$
461 T=T+1
462 IFT=11THENPRINT@299,"PROCESSING FINAL                REPORT"
463 IFT=11THENFOR F=1TO800:NEXTF:GOTO91
465 PRINT:PRINT@200,Y" + ";X;"=";
470 INPUT Z
480 IFZ=X+YTHENS=S+1:GOSUB1000:GOTO 431
485 IFZ<>X+YTHENGOSUB100:GOTO445
500 CLS:PRINT@195,"SUBTRACTION QUESTIONS"
510 PRINT:PRINT" HERE ARE 10 SUBTRACTION "
511 PRINT" QUESTIONS AT LEVEL ";A$;" TO TEST   YOUR SKILLS"
512 FORF=1TO3000:NEXTF
520 T=0:S=0
521 IFA$="L1"THENY=RND(15):X=RND(15):IFY<XTHEN521
522 IFA$="L2"THENY=RND(50):X=RND(50):IFY<XORY<15ORX<15THEN522
523 IFA$="L3"THENY=RND(200):X=RND(200):IFY<XORY<50ORX<50THEN523
524 IFA$="L4"THENY=RND(2000):X=RND(2000)
525 IFA$="L4"THENIF Y<XORY<200ORX<200THEN524
527 CLS:PRINT@76,"LEVEL"
530 PRINT@66,"SCORE":PRINT@86,"ATTEMPTS":PRINT@291,"REPORT;"
532 PRINT@99,S:PRINT@121,T:PRINT@110,A$
535 T=T+1
536 IFT=11THENPRINT@299,"PROCESSING FINAL                REPORT"
537 IFT=11THENFORF=1TO800:NEXTF:GOTO91
540 PRINT:PRINT@200,Y" - ";X;"=";
545 INPUT Z
555 IFZ<>Y-XTHENGOSUB100:GOTO527
570 IFZ=Y-XTHENS=S+1:GOSUB1000:GOTO 521
890 FORF=1TO800:NEXTF
895 GOTO 440
1000 IFS=1THENPRINT@299,"GOOD START !":FORF=1TO800:NEXTF
1005 IFS=2THENPRINT@299,"OK !":FORF=1TO800:NEXTF
1010 IFS=3THENPRINT@299,"FAIR !":FORF=1TO800:NEXTF
1015 IFS=4THENPRINT@299,"SHOWING PROMISE !":FORF=1TO800:NEXTF
1020 IFS=5THENPRINT@299,"GETTING BETTER !":FORF=1TO800:NEXTF
1030 IFS=6THENPRINT@299,"GOOD !":FORF=1TO800:NEXTF
1040 IFS=7THENPRINT@299,"VERY GOOD !":FORF=1TO800:NEXTF
1045 IFS=8THENPRINT@299,"GREAT !":FORF=1TO800:NEXTF
1050 IFS=9THENPRINT@299,"EXCELLENT !!":FORF=1TO800:NEXTF
1060 IFS=10THENPRINT@299,"FANTASTIC!!!":FORF=1TO800:NEXTF
1070 RETURN
2000 CLS: PRINT:PRINT" WHAT FUNCTION WOULD YOU LIKE"
2001 PRINT" TO TRY ;"
2010 PRINT:PRINT"        M = MULTIPLICATION "
2020 PRINT:PRINT"        D = DIVISION "
2030 PRINT:PRINT"        A = ADDITION "
2040 PRINT:PRINT"        S = SUBTRACTION "
2045 PRINT:PRINT" SELECT THEN PRESS RETURN";
2050 INPUTI$
2055 IFI$="M"ORI$="D"ORI$="A"ORI$="S"THENGOSUB4060
3000 IF I$<>"M" OR I$<>"D" OR I$<>"A" OR I$<>"S" THEN SU=1
3010 IFSU=1THEN PRINT"I DON'T UNDERSTAND ";I$
3020 IFSU=1THENSU=0: FORX=1TO700:NEXTX:GOTO2000
```

```
4060 REM LEVEL
4070 CLS:PRINT@66,"WHAT LEVEL WOULD YOU LIKE,"
4080 PRINT:PRINT"    L1 = BENCH SITTERS"
4090 PRINT:PRINT"    L2 = MIDFIELDERS"
4100 PRINT:PRINT"    L3 = OUTFIELDERS"
4110 PRINT:PRINT"    L4 = HIGH FLYERS"
4115 PRINT:PRINT" SELECT THEN PRESS RETURN";
4120  INPUT A$
4130 IFA$="L1"ORA$="L2"ORA$="L3"ORA$="L4"THENL=1 ELSEL=5
4132 IFL=5THENPRINT"I DON'T UNDERSTAND ";A$:FORF=1TO700:NEXTF
4133 IFL=5THENGOTO4070
4140 IFL=1ANDI$="M"THENGOTO15
4150 IFL=1ANDI$="D"THENGOTO 300
4160 IFL=1ANDI$="A"THENGOTO400
4170 IFL=1ANDI$="S"THENGOTO500
30000 T$="MATH'S TESTER":T=138:M=23:V=1:GOSUB30050
30010 T$="WRITTEN BY":T=203:M=16:GOSUB30050
30020 T$="BROOKE SPONG":T=266:M=9:V=5:GOSUB30050
30030 FORX=1TO1000:NEXTX:RETURN
30040 CLS
30050 FORN=1TO LEN(T$):PRINT@T,RIGHT$(T$,N);:NEXT:SOUNDM,V
30060 RETURN
```

## SHIFT LOCK UPDATE BY JOE LEON . .

SHIFT LOCK SWITCH WITH POWER ON RESET (C) J.P. LEON 1987

IC1-4013




Instead of using a mechanical push on/push off switch an electronic FLIP FLOP can be used as a switch. This circuit uses one half of a dual flip flop and a single NAND gate left over from the updated NMI project in this issue.

The P.B. SW. shown can be a small momentary switch mounted anywhere convenient or the RIGHT SHIFT KEY on the VZ 300 Keyboard can be used. If you use the latter option then the track on either side of the Right Shift Key on the Keyboard P.C.B. must be cut. Solder a thin wire to each side of the KEYPAD on the track leading to the isolated Shift lock switch. The other ends of the two wires go as shown on the circuit diagram. While you have the Keyboard apart you may decide to mount the 3mm LED as I did right in the Shift Key Cap itself, right above the "I" in the word SHIFT. It looks neat.

Connecting the Transistor to the keyboard matrix is straightforward. When built the LED will let you know when the switch is on. If the LED is on, but Shift Lock is not then try reversing the leads from the Transistor. All unused INPUTS on both IC'S must be grounded or taken to +5V. Pinouts for both IC'S are shown to help intending constructors.

DATA PROGRAM WRITTEN BY ROBERT QUINN

"DATA" is best discribed as a expandable Diskbased Database. Robert has been able to put together a very useful Database program which is full of features you would expect to find in a more expensive program.

A Database is a program written to handle and store information. It is madeup of pages as you would have in a file or folder.

In "DATA" each page of this file will hold ten records and each record can be upto thirty characters long.

"DATA" is special in that it is expandable, in as far as how many pages and records you can have in your file. If you have a VZ300 then your Datafile will have 190 records organised on 19 pages.

On the other hand if you had a 16k memory module installed, then you will have 740 records on 74 pages, which is heap's of room for your records.

"DATA" also has all the usual features of a Database you would expect to find, such as the ability to sort or edit your Datafile. You can also save/load and make a hardcopy of your information.

If you want value for money and a way to store all your information then "DATA" is the program for you.

The cost of A$20.00 makes it one of the cheapest information storage and retrieval mediums you can buy.

My version has no instruction manual as such. But there are 3 datafiles on the Disk which, when loaded, will give you instructions on how to use "DATA". This can be a problem sometimes if you get stuck in the middle of a file and need help. It's not a big problem and is easy overcome if you make a hardcopy of the instructions.

"DATA" would be one of the most useful and value packed programs available to do what is an important function of most computer system's.

I use this program all the time and find it superb to store my club records on.

CONCLUSION

Well written program. Very useful and good value.

Review written by PETER HILL (Auckland) NZ

EDITORS COMMENT :-

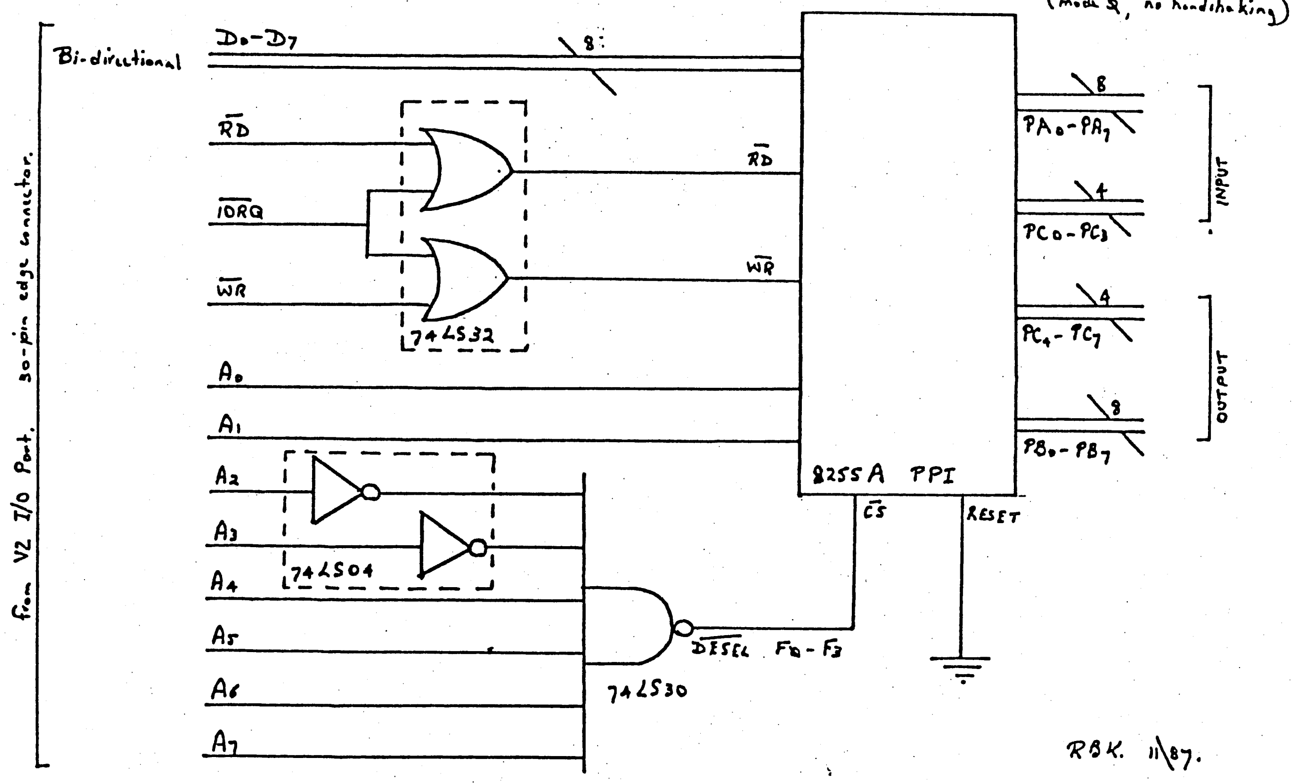
"DATA" is available in two versions as a DISK or TAPE DATABASE. In both versions fast BLOCK SAVE/LOAD techniques are used as well as fast M/C SORT and SEARCH. See VEE ZED DOWN UNDER ad at rear of Journal for more information and address where it can be purchased from. Ed.

# 24 BIT I/O I/FACE (C) R.B.KITCH 8

24 bit PROGRAMMABLE
PERIPHERAL
INTERFACE
for V2 COMPUTERS.

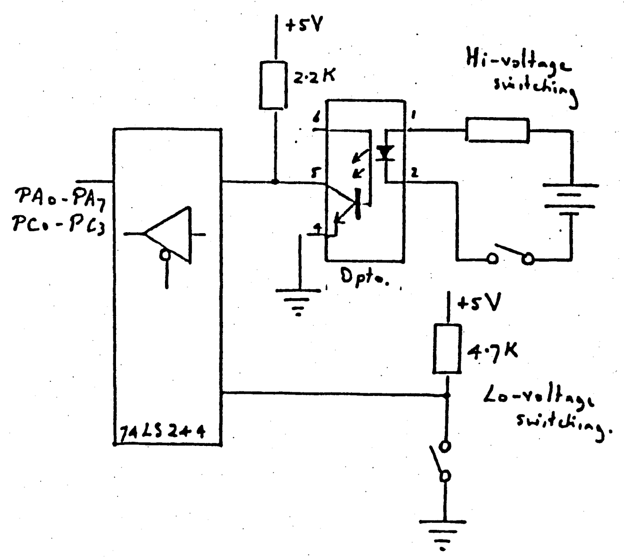
Configuration Details - as shown.

| | | | | |
|---|---|---|---|---|
| Port A | I/O addr. | F0 H | 240D | input |
| Port B | | F1 H | 241D | output |
| Port C | | F2 H | 242D | - upper output<br>- lower input. |
| Control. | | F3 H | 243D | - 91H, 145D, 10010001B<br>(mode 0, no handshaking) |



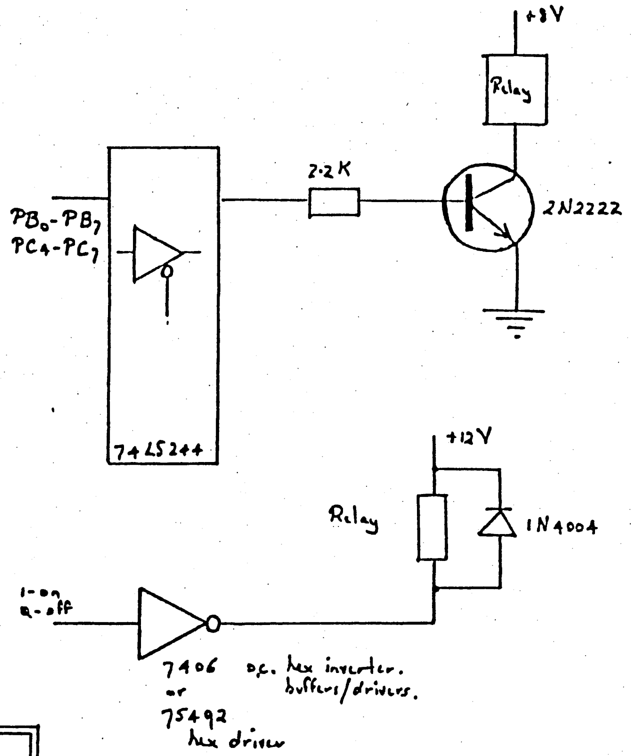

RBK. 11\87.

INPUT Interfaces.




74LS244 will handle directly 5 - 2.5 V switching.

OUTPUT Interfaces.




INTERFACING DIGITAL I/O to PPI

RBK 11\87.

# 24-BIT I/O PROGRAMMABLE PERIPHERAL INTERFACE BOARD FOR VZ200/300 COMPUTERS.

by R. B. Kitch. 7 Eurella St., KENMORE, QLD. 4068

A large number of Users require digital (switching) I/O from their VZ computer to allow interfacing to "real-world" devices. Robotic control and electric train set control are the most usually cited requirements, but house security, serial or parallel communications with other computers or printers, or synchronizing other peripherals are some possible uses of programmable ports.

The interface described herein provides, in its simplest configuration, twelve digital outputs and twelve digital inputs. The I/O lines are latched (thereby eliminating pull-up or pull-down resistors), buffered and seperately powered or isolated from the computer. The outputs may be interfaced to relays, DACs or LEDs. The inputs are usually switches or ADCs.

The interface uses a "smart" programmable port so that it can be readily reconfigured by suitable software. The Intel 82C55A PPI is used. It can also be programmed to look like three simple I/O ports (mode 0), two handshaking I/O ports (mode 1) or a bi-directional I/O port with five handshaking signals (mode 2). The modes can be mixed.

Address decoding of the four I/O ports required by the PPI on the Z80 bus structure is also provided. Decoding of the Z80's *IORQ, *WR and *RD control signals is also done on-board.

The advantage of using a "smart" interface is that the mode selection is readily selectable by one control word - no matter how complex the configuration may be. "Dumb" ports (for example 74LS244 and 74LS374 bus drivers) can however take a little more of a beating and have a higher current driving capacity. Their timing is often a little less critical than a smart port. The smart port which provides the greatest number of bits-per-dollar is the Intel 82C55A Programmable Peripheral Interface (PPI) which is the CMOS high speed version of the earlier 8255.

## FUNCTIONAL DESCRIPTION.

The 82C55A is a general purpose I/O interfacing device providing 24 I/O lines. These are organised as three 8-bit I/O ports labelled A, B and C. Individual bits cannot be programmed as inputs or outputs. Instead, all of the bits in Port A or B are programmed as one byte. The four high- and four low-order bits of Port C can be programmed as two seperate nibbles however. This compares more than favourably with the Z80 PIO device. In addition, the 82C55A does not require to be connected to the system clock, M1 and interrupt lines as required by the Z80 PIO. The 82C55A matches the terminations available from the VZ I/O port located at the rear of machine.

The PPI may be used to interface peripheral equipment onto a microcomputer bus system. The functional configuration of the PPI is programmed by the system software via the Control or Status Word port, such that usually no external logic is necessary to interface peripheral devices or structures. All communication with the PPI occurs over the eight data lines.

In fact, the PPI is actually seen as four I/O ports on the bi-directional data bus and can be mapped anywhere into address space by suitable decoding of the address lines A2-A7. The specific port address is controlled by the *CS input, which controls access to the PPI. Address lines A0-A1 determine which of the four ports is addressed as follows-

| | | | |
|---|---|---|---|
| Port A | addr. 0 | A0=0 | A1=0 |
| Port B | addr. 1 | A0=1 | A1=0 |
| Port C | addr. 2 | A0=0 | A1=1 |
| Control | addr. 3 | A0=1 | A1=1 |

When *IORD is low, any of the three data ports can be read by applying the appropriate combination to A0 and A1. When both A0 and A1 are high, the Control Port is accessed. Note that this register can only be written to, not read from. When the PPI is not accessed (*CS high or, *WR and *RD high), the data bus connections are placed in a high impedance state.

Port A can be configured as-
i/. one 8-bit data output latch/buffer with 2.5mA DC drive capacity - enough to drive one standard TTL load or four LSTTL loads.
ii/. one 8-bit data input latch/buffer.
(Both pull-up and pull-down bus hold devices are present).

Port B can be configured as-
i/. one 8-bit input data output latch/buffer with 2.5mA DC drive capacity.
ii/. one 8-bit data buffer (no latching).
(Only pull-up bus hold devices are present).

Port C can be configured as-
i/. one 8-bit data output latch/buffer with 2.5mA DC drive capacity.
ii/. one 8-bit data input buffer (no latching).
iii/. two 4-bit latched ports under mode control.

Each port contains a 4-bit latch and can be used for control signal outputs and status signal inputs in conjunction with Ports A and B. Only pull-up bus hold devices are present.

The Control Port can be both written to and read from to achieve control. Note that the "asymmetry" of the port features can be used to advantage when designing I/O requirements with respect to latching/non-latching.

INTERFACING I/O INTO THE PPI.

The PPI can sink or source 2.5mA on output. This may be enough to drive low power devices, but generally to drive LEDs (20mA) or relays (20-100mA) higher current sourcing capacity is required. VLSI devices are also not terribly immune to noise or voltage spikes. It is generally safer therefore to interpose bus drivers/buffers. In the PPI configuration detailed herein, all latching of I/O data is taken care of in the PPI. Propagation delays introduced by the 74LS-series devices are not significant to a 4MHz Z80 system - particularly when the Z80 inserts an (extra) WAIT state into the timing of any read/write operation involving an I/O port. This feature of the Z80's timing allows "slow" peripheral devices time to latch data. This timing feature is the reason why it is better to map the PPI into I/O port address space rather than memory address space.

The data supplied indicates interface circuits for high and low voltage switching via a 74LS244 bus driver. Isolation is provided where necessary by opto-isolators or relays. A wide variety of circuit configurations are possible here depending upon what is being interfaced onto the bus.

SELECTING OPERATING MODE.

Once the PPI has been interfaced between the Z80 bus system and the "real-world" devices, the operating mode must be selected by configuring the Control Port. Only mode 0, for basic I/O is detailed herein. Basic I/O uses unconditional or non-handshaking features, and is best used when the peripherals can always be assumed to be ready.

A single control word is written to the Control Port.
This word is built up as follows-
Bit 7 =1 modes 0/1/2 to be selected (rather than bit set/reset mode).
Bit &5=00 mode 0 selected (01 for mode 1, 1X for mode 2).
Bit 4 =1 Port A - input (0 for output).
Bit 3 =0 Port C upper nibble - output (1 for input).
Bit 2 =0 mode 0 selected (1 for mode 1).
Bit 1 =0 Port B - output (1 for input).
Bit 0 =1 Port C lower nibble - input (0 for output).

The control word is 10010001B, 91H or 145D for the configuration shown here. It can be seen that the three ports are broken into two groups for mode selection. Port A and the high-order bits of Port C can be programmed for any of the modes 0 through 2. Port B and the low-order bits of Port C can be programmed in modes 0 and 1 only.

## JAGULAR JETTY BY ADAM MAGEE . . .

```
2 '***********************************************************
4 '*   JAGULAR JETTY BY ADAM MAGEE - MEMBER OF H.V.VZ.U.G.   *
6 '***********************************************************
8 :
10 POKE30744,1:GOTO 9000
11 HS=999
12 FORL=3TO1STEP-1
13 CLS
14 IFL=0THENGOTO7000
15 CP=28672
16 X=1
18 COLOR2:PRINT@206,"■■■"
20 FORI%=256TO287:POKECP+I%,191:NEXT
30 FORI%=288TO383STEP2:POKECP+I%,207:NEXTI%
40 FORI%=286TO413STEP32:PRINT@I%,"  ";:NEXT
45 PRINT@32,"LIVES"L
50 A$=INKEY$
60 IFA$="M"ANDX>0THENX=X-1
70 IFA$=","ANDX<31THENX=X+1
75 YP=28896+X
80 K=PEEK(YP)
81 IFYP>28924ANDYP<28928THENGOTO6000
82 IFK=149ORK=154ORK=159THENGOTO3000
83 IFPEEK(YP+32)=207THENGOTO4000
84 IFPEEK(YP+32)=32THENGOTO5000
90 POKEOP,32:POKEYP,43
91 PRINT@20,"SCORE"SC
92 PRINT@9,"HIGH"HS
```

```
100 OP=YP
105 SC=SC+1
110 P=RND(9)
115 IFP=6THENP=9
120 IFP=1THENCOLOR4:PRINT@258,"■ ■ ■ ■ "
130 IFP=9THENCOLOR4:FORP=0TO7:POKECP+258+P,191:NEXT
140 IFP=3THENCOLOR2:PRINT@206,"    ":PRINT@238,"■■■"
150 IFP=4THENCOLOR2:PRINT@206,"■■■":PRINT@238,"    "
160 IFP=5THENGOSUB1000
170 IFP=7THENGOSUB2000
180 GOTO 50
1000 FORK=0TO3:POKECP+281+K,207:NEXTK:RETURN
2000 FORK=0TO3:POKECP+281+K,191:NEXT:RETURN
3000 CLS:PRINT
3010 PRINT" YOU'VE BEEN HIT BY THE CHOMPER"
3020 PRINT" AND SQUASHED TO DEATH":PRINT
3030 PRINT" YOU LOSE A LIFE";
3040 PRINT" LIVES LEFT"L-1
3060 FORT=15TO0STEP-1:SOUNDT,1:NEXTT
3070 NEXTL
3075 IFL=0GOTO7000
4000 CLS:PRINT:PRINT" OH NO YOU'VE TROD ON THE DEATH"
4010 PRINT" WALK .  YOU LOSE A LIFE":PRINT
4020 PRINT" LIVES LEFT"L-1
4030 FORI=1TO20:SOUNDRND(31),1:NEXTI
4035 NEXTL
4040 IFL=0THENGOTO7000
5000 CLS:PRINT:PRINT"   BUZZ Z  - STEPPING ON THAT":PRINT
5010 PRINT"   ELECTRIC PLANK REALLY GAVE":PRINT
5020 PRINT"   YOU A SHOCK - LIVES LEFT";L-1
5030 FORT=10TO31:SOUNDT,1:NEXT
5040 NEXTL
5050 GOTO7000
6000 CLS:PRINT:PRINT" YOU'VE MADE IT.  WITH"L"MEN TO"
6010 PRINT" SPARE .  THAT MEANS A BONUS OF"
6020 PRINTL*10;"SO YOUR FINAL SCORE WAS "SC-L*10:SC=SC-L*10
6030 IFSC<HSTHENPRINT" WHICH IS THE NEW HIGH SCORE!!!"
6035 IFSC<HSTHENINPUT" WHAT IS YOUR NAME CHAMP . . .   ";C$
6040 IFSC<HSTHENHS=SC
6050 PRINT:PRINT" HIGH SCORE OF";HS;"WAS OBTAINED"
6060 PRINT" BY ";C$
6065 SOUND28,2;23,1;23,1;24,2;23,2;0,1;27,2;28,2
6070 PRINT:PRINT" ANOTHER GAME ";C$;
6075 PRINT" Y/N ";:SC=0
6080 INPUTM$:IFM$="Y"THENGOTO12ELSEEND
7000 CLS:PRINT@75,"GAME OVER!"
7010 PRINT:PRINT" SCORE"SC
7020 GOTO6050
9000 CLS:B$=" JAGULAR JETTY ! JAGULAR JETTY !"
9010 FORB=1TOLEN(B$)
9020 PRINT@0,LEFT$(B$,B):PRINT@32,RIGHT$(B$,B)
9030 PRINT@64,LEFT$(B$,B):NEXTB
9040 PRINTTAB(10)"USING KEYS"
9050 PRINT:PRINTTAB(5) "(M) LEFT -- (,) RIGHT"
9060 PRINT
9070 PRINT" GUIDE YOUR WAY ALONG THE JETTY"
9080 PRINT" AVOIDING THE ELECTRIC PLANKS ,"
9090 PRINT" FISH PROCESSORS ,   AND MOVING"
9100 PRINT" WALKWAYS .   THE FASTER ACROSS"
9110 PRINT" THE JETTY THE BETTER THE SCORE"
9120 PRINT:INPUT"  HIT <<RETURN>> TO START ";X9
9130 GOTO 11
```

BANK SWITCHED MEMORY IN THE 64K EXPANSION PACK by Chris Hobrough

The 64K expansion module for the VZ 200 and 300 performs two functions. Firstly, it fills out the top half of the memory range that the Z-80 processor is capable of addressing directly and secondly, it provides two additional banks, of 16K each, which can be switched in when necessary.

The module is divided into four 16K banks numbered 0 to 3. Bank 0 occupies the area from 32K to 48K and is fixed. Banks 1 to 3 all effectively occupy the area from 48K to 64K and are switchable. In other words, for each logical address in this range there are three physical locations. Needless to say, only one of the three is available at a time. On power up the default setting is with bank 1 enabled and banks 2 and 3 disabled.

The leaflet that comes with the module claims that you cannot use this bank switching feature from BASIC, but all this means is that the interpreter cannot make use of it directly. If you are programming in BASIC then just lower the Top of Memory below the 48K point (see my article on saving memory to tape if you are unfamiliar with the use of the memory pointers) and this will leave the top 16K free to play with as you wish without upsetting the BASIC interpreter.

The software switch for the memory banks occupies the I/O address range 70-7FH (or 112-127 decimal. NB. this is not the same as a memory address) and is operated simply by writing the number of the bank you wish to select to an address in that range with an output command. In BASIC this would be:-

```
OUT 127, N
```

Where "N" is the number of the bank you want (1-3). The "127" could have been any number from 112 to 127, it doesn't matter. If you are using assembly language then first load the bank number into the A register and then use the OUT(N),A instruction :-

```
LD   A, N     OUT  (7FH), A
```

Well, that takes care of the HOW but leaves you with the WHEN and WHY. This is really wide open and depends on your programming experience and imagination. One thing is certain, you can't just write away in BASIC and hope for the extra memory to be switched in when needed. You have to do it yourself and it requires careful planning.

Swapping banks for one memory access is possible but rather slow and so the most practical use is for machine code sub-routines or for blocks of data. Remember that any routines which are needed all the time or which call another bank must be below the 48K point or else they will become unavailable when you switch.

When using the extra memory for data storage from BASIC it is possible to use BASIC's array structures by leaving the Top of Memory at 64K and just lowering the stack below 48K. However, this is very complicated and arrays use a lot of memory for overheads, which further cuts down the space for your main program. It is best to POKE the data into known positions directly. Data base records, for instance, are often of fixed

length and contain individual data fields of predetermined size. You only need to know the address where the file begins in order to calculate the position of any field in any record as required.

Finally, I'll leave you with one thought. With the help of a short machine code routine a Hi-Res screen can be copied from anywhere in memory into Video RAM in the blink of an eye. Between them the three switchable banks could store 24 complete screen images. A bit of imagination could produce some fairly spectacular animation.

****************************************************************

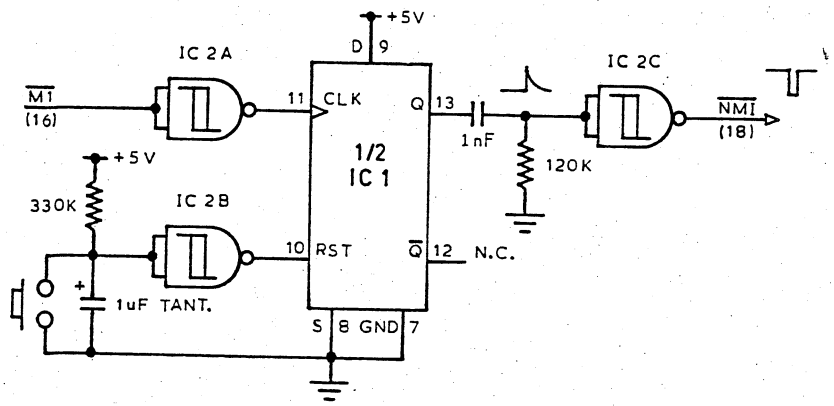
The following is a simple monitor type program to illustrate one way to access bank switched RAM. It allows you to select a bank and then enter text or view the content of the bank. Try entering different things in each bank at the same address.

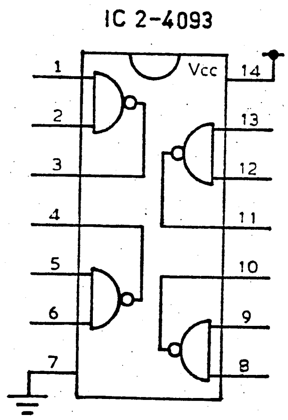
```
10 '*********************************************************
20 '*   MEMORY BANK MONITOR BY C.J. HOBROUGH - NOVEMBER 1987  *
30 '*********************************************************
40 :
99 'LOWER TOP OF MEMORY TO BELOW 48K *************************
100 POKE30898,191
110 CLEAR100
999 'MAIN MENU ********************************************
1000 CLS
1010 PRINT@34,"BANK SWITCHED MEMORY MONITOR"
1020 PRINT@66,"----------------------------"
1030 PRINT@135,"STORE DATA IN BANK"
1040 PRINT@207,"OR"
1050 PRINT@260,"EXAMINE CONTENTS OF BANK"
1060 A$=INKEY$:A$=INKEY$
1070 IFA$<>"S"ANDA$<>"E"THEN1060
1080 IFA$="S"GOSUB2000            'STORE
1090 IFA$="E"GOSUB3000            'EXAMINE
1100 GOTO1000
1999 'STORE ***********************************************
2000 GOSUB4000                    'SELECT BANK & ADDRESS
2010 CLS
2020 PRINT"TYPE IN CHARACTERS TO BE STORED"
2030 PRINT"AND PRESS <RETURN> TO FINISH."
2050 PRINT
2100 PRINT"■";:PRINTCHR$(8);     'PRINT CURSOR & BACKSPACE
2110 A$=INKEY$:A$=INKEY$         'WAIT FOR INPUT
2120 IFA$=""THEN2110             'BACK IF NONE
2130 POKE30862,80:POKE30863,52   'SET USR POINTER
2140 X%=USR(0)                   '& CALL BEEP ROUTINE
2150 IFA$=CHR$(13)THENRETURN     '<RETURN> - EXIT TO MAIN MENU
2160 IFASC(A$)<32ORASC(A$)>95THEN2110
2161                             'CHECK FOR TEXT ONLY
2170 POKEAD,ASC(A$)              'POKE ASCII CODE INTO MEMORY
2180 AD=AD+1                     'NEXT MEMORY ADDRESS
2190 PRINTA$;                    'ECHO TO SCREEN
2200 FORT=1TO50:NEXT             'DELAY
2210 GOTO2100                    'BACK FOR NEXT CHARACTER
2999 'EXAMINE *********************************************
3000 GOSUB4000                   'SELECT BANK & ADDRESS
3010 CLS
3020 PRINT"PRESS <SPACE> FOR NEXT LINE OF"
3030 PRINT"CHARACTERS OR <RETURN> TO EXIT."
```

```
3040 PRINT
3100 FORX=1TO32
3110   PK%=PEEK(AD)                    'FETCH CHARACTER CODE
3120   IFPK%<32THENPK%=95              'CHECK FOR PRINTABLE ASCII CODE
3130   PRINTCHR$(PK%);                 'PRINT CHARACTER
3140   AD=AD+1                         'NEXT MEMORY ADDRESS
3150 NEXT                              'REPEAT FOR LINE OF TEXT
3200 A$=INKEY$:A$=INKEY$               'WAIT FOR <SPACE> OR <RETURN>
3210 IFA$=CHR$(32)THEN3100             '<SPACE> - NEXT LINE OF TEXT
3220 IFA$=CHR$(13)THENRETURN           '<RETURN> - EXIT TO MAIN MENU
3230 GOTO3200                          'BACK IF NEITHER
3999 'SELECT BANK & ADDRESS **********************************
4000 CLS
4010 PRINT@33,"SELECT MEMORY BANK (1-3)";
4020 INPUTBK%                          'SELECT BANK
4030 OUT127,BK%                        '& SWITCH IT IN
4040 PRINT@97,"SELECT STARTING ADDRESS:-"
4050 PRINTTAB(2);"(49152-65535)";
4060 INPUTAD                           'SELECT ADDRESS,
4070 IFAD>65535ORAD<49152THEN4040
4071                                   'CHECK RANGE
4080 AD=AD-65536                       '& CONVERT IT TO LEGAL ADDRESS
4081                                   'FOR POKE COMMAND
4090 RETURN
```

## NMI UPDATE BY JOE LEON . . . . . .

SYNCHRONIZED NMI AND RESET



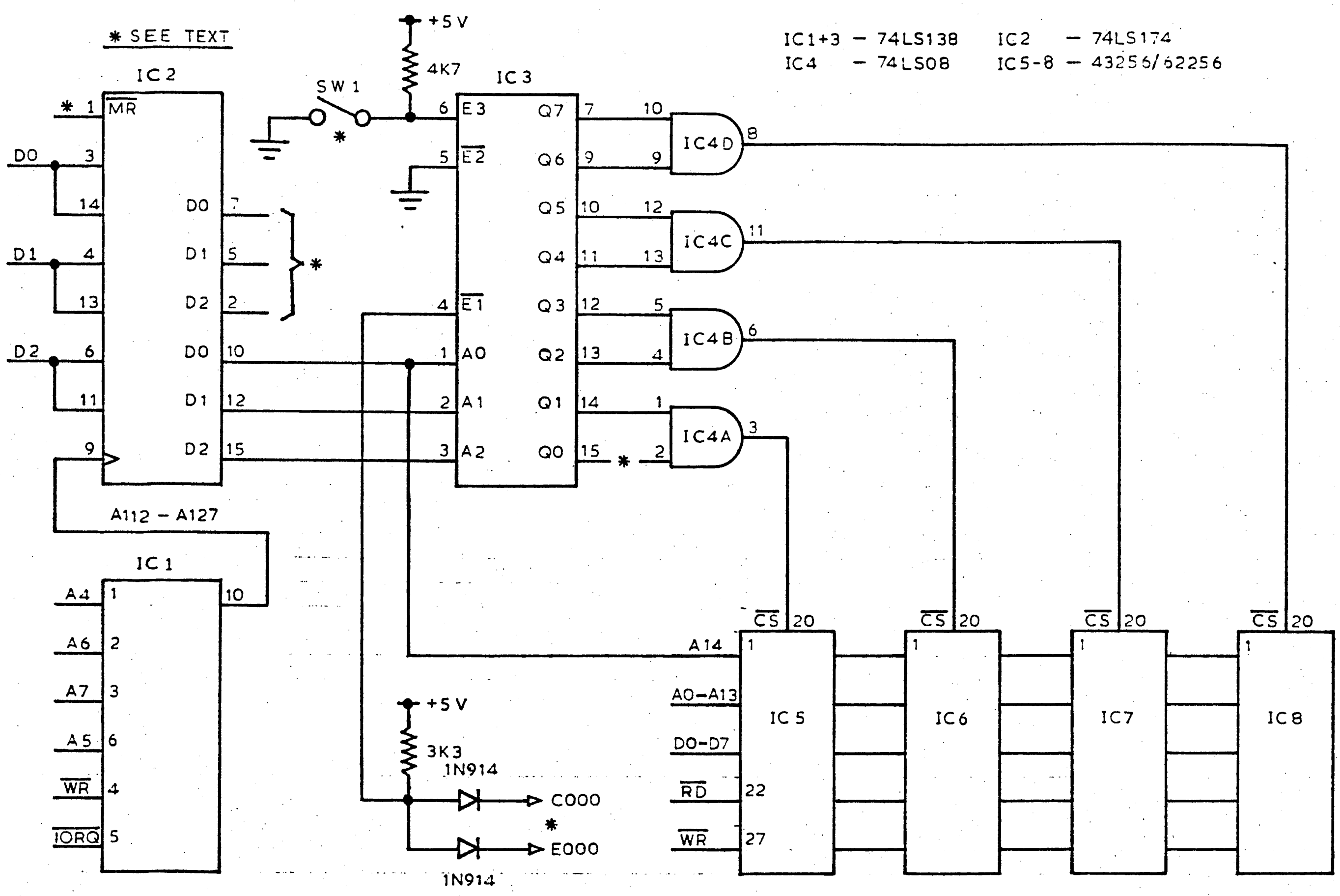

The NMI I/P is triggered on the rising edge and the MI O/P goes low at the START of a MEMORY CYCLE and this condition is used to achieve a NMI or a RESET which is SYNCHRONIZED to the start of a memory cycle. A Reset pulse of 1mS or greater will destroy Dynamic Ram and for that reason a very short pulse from IC2C is produced.

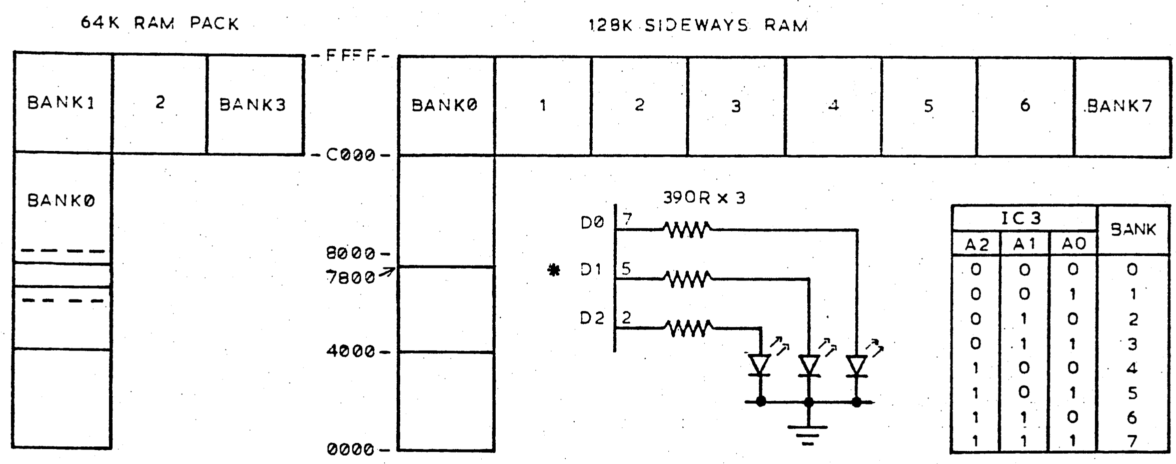
When the P/Button is pressed the F/F is reset by a positive pulse and nothing more happens till the P/B is released. Upon release the MI (Pin 16 Edge Connector) output from the Z80 triggers the F/F.

The Q O/P goes high and the positive RC edge detector supplies a 96uS pulse to IC2C which delivers a brief negative pulse to the NMI (Pin 19 on Edge Connector).

Hold CONTROL down prior to pressing NMI button for a SYNCHRONIZED RESET.

VZ 200/300 128K SIDEWAYS RAM (C) JOSEPH P. LEON 1988




| IC3 A2 | IC3 A1 | IC3 A0 | BANK |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |

NOTE ON IC'S 1 to 4 :-

Although 74LS Series IC'S are shown in above circuit and they will work as designed but if possible get for :-

IC1-74HCT138 IC2-74HCT174 IC3-74HC138 IC4-74HC08

The 74HCT Series are designed to interface between TTL and CMOS IC'S while the 74HC Series can drive TTL or CMOS IC'S. The HC/HCT IC'S have much lower power cosumption than 74LS IC'S, comparable speed and are preferable. The VZ power supply has its limits and every IC added puts an added load on the supply.

VZ 64K memory cartridges are very hard if not impossible to get. For that reason I decided to design my own and this 128K SIDEWAYS RAM project is the result and it will give you more memory than you'll ever use.

Not all of you will need the full 128K and for that reason it was designed to start with 32K and expand in 32K steps as funds allow or need arises. And now to the circuit description. Construction details will be given in next issue.

We'll start with a look at the 64K RAM PACK. Of course as you may have realised by now the VZ memory map does not allow for more than 34K of user RAM, so BANK SWITCHING techniques must be used. The block diagram of 64K Ram Pack says it all. Each of the four banks consists of 16K blocks. Bank 0 is fixed between 32K and 48K while bank 1 resides at 48K to 64K and is the default at power up. Banks 2 and 3 can be switched in when desired.

Now if you have a look at 128K S/Ways Ram block diagram you'll notice that all 8 banks from 0 to 7 reside between 48K and 64K with only one bank available at any one time. By adding the 18K of Ram below 48K we get a total of 146K ram available to the user which puts a few big brand name computers to shame. And now to the circuit proper.

IC 1) The 74LS138 provides a decoded output in the address range of 112 to 127 which is used to trigger the latch.

IC 2) The 74LS174 is a HEX LATCH and was chosen so duplicate outputs could be obtained. One lot of outputs is used to select desired bank while the second lot of outputs are used to drive LEDS to indicate bank selected.

IC 3) This 74LS138 is used to decode banks 0 to 7.

IC 4) The 74LS08 is a quad and gate and each gate is used to enable each 32K Ram twice. The ZERO DATA line which goes to each A14 on the 32K RAMS is used to enable top or bottom 16K block on each 32K Ram chip.

IC 5-8) These chips are 32K X 8 Static Cmos Ram Memory Chips. They come under two different numbers - 43256 or 62256. Both are same exept they are made by different companies.

To understand the operation of this circuit as a whole you have to know or learn a bit about the BINARY system and digital logic. Refer to the truth table for IC 3.

* Pin 1 MR IC2 - This pin is connected to pin 2 (Reset) of edge connector. This has the effect of synchronized RESET of 128K Ram and VZ at power up.

* SW1 - This switch serves two functions.
1) Switch CLOSED - All O/P's on IC 3 are forced High which in turn puts a High on each O/P of IC 4 disabling all 32K Ram IC'S thus allowing WORDPRO cartridge operation.
2) Switch CLOSED before power up then OPENED after power up. This has the effect of lowering DOS communication region and T.O.M. below 48K and very usefull with disk drives.

* D0, D1 & D2 IC2 - These O/P's are used to drive LEDS to indicate selected bank and are optional.

* C000 & E000 - These I/P's require decoded 8K blocks in the range C000-FFFF and are used to enable the 32K Ram chips via IC3 & IC4. More on them in next issue.

* Q0 Pin 15 IC3 - As it stands the 128K S/Ways Ram is not compatible with existing 64K programs using bank switching because bank 0 is at 48K to 64K. The circuit changes (Minor) will be given next issue to make it compatible with the 64K Ram Pack.

EXTENDED DOS V1.3 - $15.00

The previous version has been updated with extra commands added.

COMMANDS :-

MERGE - MERGES basic file from disk with file in memory.
DIRA - Prints DIRectory in 3 columns across the screen.
LDIRA - As above, but to screen and printer.
DIRB - Prints Track, Sector No, Start, End, Length of program.
LDIRB - As above, but to screen and printer.
OLD - Restores a program after using the NEW cqmmand.
OLD. - Prints START, END and LENGHT of BASIC file in memory.
DEC - Converts DECIMAL to HEX
HEX - Converts HEX to DECIMAL

STATUSA - Prints free disk space to screen on one line.
LSTATUSA - As above, but to screen and printer.
STATUSA and LSTATUSA also works with Version 1.0 DOS.

NEW COMMANDS :-

MENU - Loads and RUNs Binary or Text MENU program from disk.
CODE - Simplifies using printer control codes directly or from within a program.
LTAB - Is for setting of Left Margin.
MOVE - Moves Basic file from disk to chosen memory address.
UPD - Erases old file and saves with same file name.

MENU/FILE COPIER/DISK ORGANISER - $15.00

This utility will read your disk directory and present you with several options. Using the Cursor you can RUN/BRUN any program or select FILE COPY, REN, ERASE, DRIVE 1 OR 2, etc. Besides COPYING TEXT and BINARY files all other files can be copied as well exept for DATA files.

DISKFILER - To be released soon

This utility will allow you to keep track of what files are on which disk with Printout and many other options. It's a Database for your disk files.

For purchase or info contact - Dave MITCHELL - (079) 27 8519
24 ELPHINSTONE St. NORTH ROCKHAMPTON QUEENSLAND 4701

For information or demonstration in Newcastle area contact :-
Joe LEON - (049) 51 2756 - 22 DRURY St. WALLSEND NSW 2287

## MATTHEW TAYLOR SOFTWARE . . . . .

VZ SCREEN ED - $25.00 + $2.00 P&H.

VZ Screen Ed is a new drawing program for HI-RES screens with many powerful options. Both tape and disk users are catered for.

Send your Cheque or Money Order to :-
MATTHEW TAYLOR SOFTWARE -- OR PHONE (049) 75 2350
38 FISHING POINT ROAD RATHMINES NSW 2283 Australia

You can see Matthew at club meetings for a demo of Screen ED.

# VZ DOWN UNDER

## SOFTWARE

We have a range of software available at reasonable prices.

***GRAFSTAR*** This utility is designed as a tool to implement graphic display screens in BASIC programs (or on their own) without the guesswork and great amount of time that would be necessary without *GRAFSTAR*. *GRAFSTAR* uses machine code routines to do most of the usual time consuming operations, so it is FAST to use. Once loaded, you can use *GRAFSTAR* to draw your artwork with its many editing features. Once you are satisfied with your design, you can store it in memory in the blink of an eye. Screens can be saved to both tape and disk for later retrieval. But most importantly, *GRAFSTAR*, when instructed, creates a DATABLOCK, which consists of a basic program (about 5 lines) and a machine code routine which resides at the top of available memory. You can now type in your basic program, or merge an existing one, in below it. The routine, called with just GOSUB65526 switches to MODE(1) and prints your artwork onto the screen in just a fraction of a second.

This incredible program is an absolute must for the serious graphics programmer and is available for only **$20**

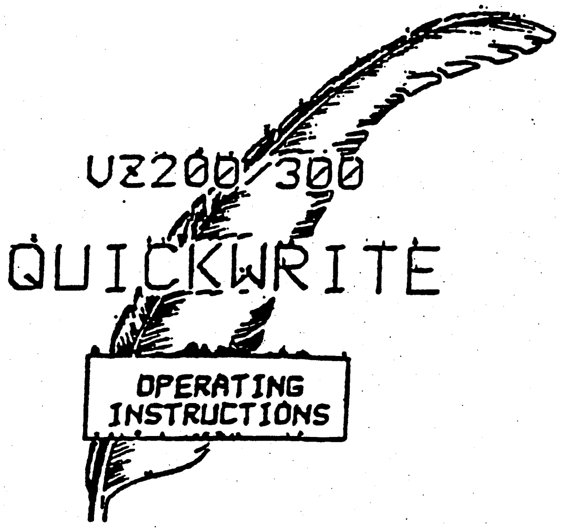
***DATA*** Ever had the need for a database that does just about everything? Not just a mailing list or disk cataloguer, *DATA* can be taylored to suit your very needs. A few of the features include automatic incrementing for fast data entry, numerous search & edit functions to make using *DATA* friendly. Disk *DATA* has DIRectory and ERAse commands aswell as a DIRectory copying feature which allows the directory of a disk to be copied as data, and *DATA* saves it to disk as a single B file. Cassette *DATA* saves datafiles as a single T file..no slow loading of multitudes of D files.

Available on both tape and disk for only **$20**

Write for a free comprehensive list of available software.

# VZ DOWN UNDER

59 Brentwood Dve Wantirna Vic 3152




*** QUICKWRITE WORDPROCESSOR ***

NEW VERSION V4.

Version V4 does not replace Version V3. V4 is a little more complicated to use but has more facilities.

The main one being that printer print styles, often wrongly called fonts, can be changed anywhere in the data/text. This means even part of a word.

*Another feature is the ability to* scroll up and down, to start or end of the text. This is rather small to read but it demonstrates *what can be done.*

This is printed by a CITIZEN 120 D printer which can print *in the* mode called inversed.

**Price A$40.00.**

Only available from VSOFTWAREZ
39 Agnes St., TOOWONG. QLD. 4066.
AUSTRALIA.
Phone (07) 371 3707.

## E & F PATCH 3.1 (C) H.V.VZ.U.G.

This single Patch will convert your E & F TAPE WORD PROCESSOR for full DISK use while retaining all TAPE functions. It can be used with 1 or 2 DRIVES. Below are the two Menus.

| | |
|---|---|
| E)DIT TEXT | L)OAD |
| C)LEAR TEXT | S)AVE |
| P)RINT TEXT | D)IR |
| L)OAD FILE | E)RA |
| S)AVE FILE | R)EN |
| V)ERIFY FILE | I)NIT |
| Q)UIT PROGRAM | 1-2) DRIVE 1 |
| D)ISK | M)ENU |

Fast SAVING and LOADING of TEXT DATA to and from Disk is provided using Block SAVE/LOAD techniques. Full instructions are supplied together with a Tape to Disk transfer utility for your E & F Tape Word Processor.

This Patch will work with V1.0 or V1.2 Disk Controller. A STATUS facility has been added for V1.0 DOS owners.

SYSTEM REQUIREMENTS :- DISK DRIVE + V1.0 OR V1.2 DOS
VZ300 + 16K RAM PACK OR
VZ200 + 18K (16K RAM PACK + 2K)

The price - $13.00, NZ AU$15.00 and is available from :-
HUNTER VALLEY VZ USERS' GROUP P.O.BOX 161 JESMOND 2299
N.S.W. AUSTRALIA Phone (049)51 2756